Amendments to the Specification

Please replace the paragraph on Page 1, lines 4 - 8 with the following marked-up replacement paragraph:

-- The present invention is related to the following commonly-assigned inventions, which were filed concurrently herewith: U. S. ~~Patent~~ ________ ~~(serial~~ Patent Application serial number ~~10/______), titled~~ 10/733,985, titled "Custom Subscription Builder" and U. S. ~~Patent~~ __________ ~~(serial~~ Patent Application serial number 10/734,043, titled ~~10/______), titled~~ "Intelligent Data Query Builder". These co-pending applications are hereby incorporated herein by reference. --

Please replace the paragraph that begins on Page 2, line 14 and carries over to Page 3, line 5 with the following marked-up replacement paragraph:

-- There are several drawbacks to existing techniques. If a content provider has not provided a subscription interface, then users are unable to subscribe to the content. Instead, they have to repeatedly take explicit action to continue viewing the content, such as returning to the content provider's Web page periodically. Another drawback of existing techniques is that, even if a subscription interface has been provided, it often does not [[has]] have sufficient granularity to meet the needs or desires of end users. As a result, the user effectively gets no say as to what data feed is important to him/her. Furthermore, users may be unaware that they might benefit from a subscription to content. For example, novice users may not realize that a subscription-type function exists. As another example, users may not take time to notice that they repeatedly visit a particular content location to explicitly request updated content. Disadvantages are therefore realized not only by end users, but also by the content providers, who thereby fail to maximize exposure of their provided content (and as a result, may miss out on potential revenue). --

Please replace the paragraph that begins on Page 3, line 17 and carries over to Page 4, line 2 with the following marked-up replacement paragraph:

-- Still another object of the present invention is to provide techniques with which content providers can locate end users who are candidates for subscribing to the provider's content[[.]], and provide those users with an easy-to-use subscription interface. --

Please replace the paragraph on Page 9, lines 2 - 9 with the following marked-up replacement paragraph:

-- The related U. S. ~~Patent ________ (serial~~ Patent Application serial number 10/734,043, titled ~~10/______), titled~~ "Custom Subscription Builder", referred to hereinafter as "the first related invention", discloses techniques whereby a user may subscribe to content even though the content provider has not deployed a subscription interface. Embodiments of the present invention may be used to offer subscriptions to users for content which does have an existing subscription interface, or for content which does not. In cases where a subscription interface does already exist, the interface disclosed in the first related invention is preferably used instead. Thus, references hereinafter are in terms of that interface. --

Please replace the paragraph on that begins on Page 14, line 17 and carries over to Page 15, line 8 with the following marked-up replacement paragraph:

-- Referring again to Fig. 15, reference number [[1502a]] 1502b indicates that the candidate subscription information is forwarded to a component referred to herein as CP2XML 1530, as stated above. The CP2XML component receives the forwarded serialized information, along with an indication of the source (e.g., the URL) from which the content was originally clipped. The CP2XML component then transcodes the serialized information. Preferably, the output of the transcoding operation is an Extensible Markup Language ("XML") document. (Note that if a clipping component is not used in a particular implementation of the present invention, the HTML content to be transcoded may be created using other means, including a simple text editor, and forwarded to the CP2XML component for transcoding. As another alternative, the XPath statements illustrated by the example in Fig. 3 may be written using other means, including a text editor or other tool, and the XPath statements can then be executed against the source file to generate the HTML content to be delivered to the CP2XML component.) --

Please replace the paragraph on Page 15, lines 9 - 19 with the following marked-up replacement paragraph:

-- Operation of the CP2XML component will now be described in more detail with reference to Figs. 4 - 8 and 15. Fig. 4 provides a sample subscription page 400 that may be offered to the candidate user for establishing a content subscription. See the welcome message at reference number 415, which includes a description of the candidate content (in this case, by specifying the content's URL at 416; alternatively, a description may be provided in another form, such as by using a graphical image or icon). In this example, the sample page 400 also enables the user to customize his/her subscription by specifying conditions that must occur before the user is interested in receiving a content update. In preferred embodiments, this page is built from the HTML input provided by the [[Web]] clipping ~~component~~ engine at 1502a. Reference number 1503 of Fig. 15 represents the CP2XML component 1530 creating and rendering this candidate subscription page 400 to the user. --

Please replace the paragraph on Page 17, lines 7 - 12 with the following marked-up replacement paragraph:

-- Reference numbers 541 and 542 in Fig. 5 indicate that the user interacts with this subscription page. For example, if the user decides to accept the suggested subscription, then he/she may proceed to customize that subscription, and submit the customization when ready (as discussed below with reference to Fig. 4). The CP2XML component then receives the customization input (Block [[540]] 550), and passes (Block [[550]] 560) the URL of the source Web page and the customization information to a trigger handler component. --

Please replace the paragraph on Page 23, lines 11 - 16 with the following marked-up replacement paragraph:

-- As stated previously with reference to Block 560 of Fig. 5, the customization information is sent by the CP2XML component to a trigger handler component. This passing of information is also depicted at reference number 1504 of Fig. 15, and the trigger handler is shown at reference number 1540. In preferred embodiments, the information passed to the trigger handler by the CP2XML component is encoded as an XML document. A sample document for the ~~job postings~~ online shopping scenario, corresponding to the customizations illustrated in Figs.

4 and 7, is presented in Fig. 8. --

Please replace the paragraph that begins on Page 23, line 17 and carries over to Page 24, line 8 with the following marked-up replacement paragraph:

-- The trigger handler 1540 of preferred embodiments transforms the XML document received from the CP2XML component into a trigger that a content matching engine understands. Preferably, a commercially-available content matching engine which operates in a publish/subscribe mode is leveraged, and the transformation performed by the trigger handler component comprises adapting the XML document to the application programming interface ("API") used by that content matching engine. The content matching engine is depicted in Fig. 15 at reference number 1570, and the passing of the adapted XML document from the trigger handler to the content matching engine is shown at 1505. (The format of the XML document passed to the content matching engine ~~which~~ will vary, depending on the API of the particular content matching engine which is deployed with an implementation of the present invention, and this document has therefore not been illustrated.) --

Please replace the paragraph on Page 30, lines 1 - 6 with the following marked-up replacement paragraph:

-- As has been described, embodiments of the present invention provide a number of advantages to end users and to companies. An implementation of the present invention may be offered as a stand-alone product or as a service, or it may be coupled or integrated with another software product such as IBM's WebSphere® Everyplace® Access or IBM's INS product.

(“WebSphere” and “Everyplace” are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.) --